CS2010 PS5 - The Onset of Labor v4

Released: Sunday, 12 October 2014, 8pm
Due: Saturday, 18 October 2014, 8am

Collaboration Policy. You are encouraged to work with other students or teaching staffs (inside or outside this module) on solving this problem set. However, you **must** write Java code **by yourself**. In addition, when you write your Java code, you **must** list the names of every collaborator, that is, every other person that you talked to about the problem (even if you only discussed it briefly). This list may include certain posts in CS2010 Facebook group. If you have access to your seniors' CS2010 files (that is, problem sets version 1/2/3), please refrain from looking at their code verbatim. Any deviation from this policy will be considered as cheating. If the offender is caught beyond reasonable doubt, he/she will be punished severely, including referral to the NUS Board of Discipline. It is not worth it to cheat just to get 19% when you will lose out in the other 81%.

R-option. This PS has R-option at the back. This additional task usually requires understanding of data structure or algorithm *beyond* CS2010. A self research on those relevant additional topics will be needed but some pointers will be given. CS2010R students *have to* attempt this R-option. CS2010 students can choose to attempt this R-option too for extra challenge, or simply leave it.

2011 Story. After ≈ 40 weeks, ≈ 9+ months, and 3 trimesters, a baby is born. Jane's EDD (Expected Date of Delivery) was 24 October 2011 (NUS Week 11 in S1 AY2011/2012). But as many parents will testify, babies rarely born *precisely* on their EDD, but they can be born +/- 2 weeks from the EDD.

When you seniors first read this PS (release date: 4 October 2011), Jane was actually just three weeks away from birth (she was born on 24 October 2011, exactly on her EDD). By solving this PS, your seniors had a chance to help their lecturer (me =D) did something super important in his life: Accompanied his wife during *the onset of labor*[1], escorted her from home to our chosen hospital as fast as possible, so that she went through the 'challenging' three stages of labor[2] in a proper delivery suite in the hospital.

Now, it is your chance (CS2010 S1 AY2014/2015) students to do the same =D.

2014 Story. Joshua's EDD was 29 July 2014. However, Joshua was taken out to the world on 16 July 2014 (about two weeks earlier than EDD) using elective Caesarean delivery. This year, the timings are known so there is no need for Steven and Grace to rush from home to hospital like in PS5 story... However, that gives birth to PS5R story :).

The Actual Problem. Given a map of Singapore (as a directed weighted graph), estimated time[3] to travel through Singapore roads (as *non-negative* weights of the corresponding directed edges – in minutes), Steven and Grace's home (always vertex 0 on that graph), their chosen hospital (always vertex 1 on that graph), determine the *shortest path* to go from Steven and Grace's home to their

[1] The cues that birth is imminent, like the ruptured water bag, contractions, etc.
For more details, see: `http://www.babycenter.com/0_signs-of-labor_181.bc`

[2] See PS2 story: 'Scheduling Deliveries'. Now imagine that Steven's wife, Grace, is now really one of the women in the delivery room.

[3] To simplify this problem, let's assume that this time estimation is accurate and there is no traffic jam in Singapore.

chosen hospital and report the shortest path weight: The sum of edge weights along the shortest path. Steven will call for taxi and then instruct the taxi driver to take this path. It is guaranteed that there will be at least one path from vertex 0 to vertex 1 in the given graph.

Steven and Grace needs the shortest path/quickest way, because once the contractions start (**the onset of labor**), Grace does not want to spend *too much time* on the road because – as you may have guessed it correctly – it is 'not without pain'... Similarly, Steven does not want to be *overly anxious* and wants to have Grace taken care by the doctor and nurses/midwives *as soon as possible*. The first stage of labor usually last for *a few hours*.

Now, let's go back to the problem. For example, suppose Singapore[4] a directed weighted graph as shown below (all edges are directed, except edge 2-7 and 4-6 are bidirectional):

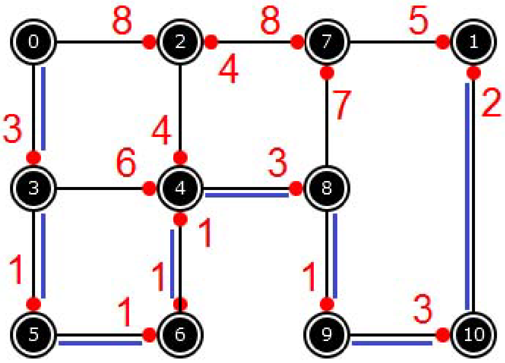

Figure 1: A Sample Singapore Map

If Steven and Grace's home is at vertex 0 and their chosen hospital is at vertex 1, then the quickest way is this path: `0 -> 3 -> 5 -> 6 -> 4 -> 8 -> 9 -> 10 -> 1` with total estimated traveling time of: 3+1+1+1+3+1+3+2 = 15 minutes. The skeleton program `Labor.java` is already written for you, you just need to implement one (or more) method(s)/function(s):

- `int Query()`
  Query your Adjacency List data structure[5] where the weight (in minutes) of each road (edge) is stored in the Adjacency List itself, and return the shortest path weight (in minutes) from vertex 0 to vertex 1. There will be at least one path from vertex 0 to vertex 1 in the given graph. The weight of each edge is non-negative and at most 1000.
- If needed, you can write additional helper methods/functions to simplify your code.

Subtask A (25 points). On an 'impossible case' that simplifies this problem, the road network in Singapore is a directed weighted tree and $1 \leq V + E \leq 10$.

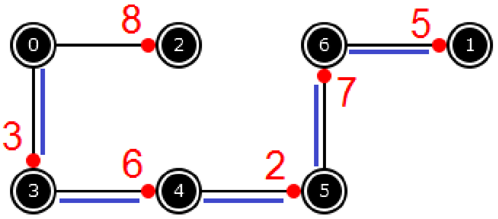

Figure 2: A Simplified Singapore Map (Tree)

The quickest way for the tree above is path: `0 -> 3 -> 4 -> 5 -> 6 -> 1` with total estimated traveling time of: 3+6+2+7+5 = 23 minutes.

[4] Yes, Singapore map does not looks like this, but let's just assume it is.

[5] Already implemented in `Labor.java`.

Subtask B (Additional 25 points). On another 'impossible case' that simplifies this problem, all roads in Singapore is a directed weighted graph but somehow require exactly 7 minutes to traverse and $1 \le V + E \le 100$.

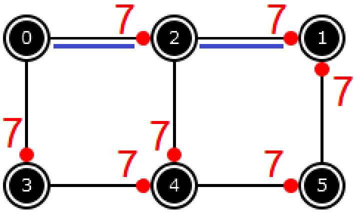

Figure 3: A Simplified Singapore Map (Similar Weight)

The quickest way for the graph above is path: `0` → `2` → `1` with total estimated traveling time of: 7+7 = 14 minutes.

Subtask C (Additional 25 points). The road network in Singapore is a directed weighted graph, the time to traverse Singapore roads varies, and $1 \le V \times E \le 1000000$. In fact, the sample test case shown in Figure 1 fits this description.

Subtask D (Additional 25 points). Same as Subtask C, but now $1 \le V + E \le 250000$.

R-option/Subtask E (No bonus point for non CS2010R students). Due to Caesarean section, Grace has to stay in the hospital for several days. Because Grace stays in a 2-bedded room, Steven cannot sleep overnight at the hospital and has to go back home. Therefore, in those days, Steven commute back and forth from home to hospital and vice versa (still using taxi). This time, to minimize boredom, he instructs the taxi driver to always take **different** paths during those commuting trips. Specifically, for this Subtask E, you need to answer a new query:

- `int Query()`
  Query your Adjacency List data structure where the weight (in minutes) of each road (edge) is stored in the Adjacency List itself, and return the **maximum number of different paths** from vertex 0 to vertex 1. There will be at least one path from vertex 0 to vertex 1 in the given graph. For this Subtask E, you can ignore the weight of each edge (they are not used) and **it is guaranteed that** $1 \le V \le 150$. Two paths that start from vertex 0 to vertex 1 are said to be different if they do not share *any edge* (the paths can share common vertices).

For example, the answer of this new `Query()` on the graph shown in Figure 1 is 2 and those 2 different paths[6] are:

1. 0 → 2 → 7 → 1
2. 0 → 3 → 5 → 6 → 4 → 8 → 9 → 10 → 1, or
   0 → 3 → 4 → 8 → 9 → 10 → 1 (notice that **all edges** are different compared to path #1)

Another example: The answer of this new `Query()` on the graph shown in Figure 2 is 1 and that only 1 path is:

1. 0 → 3 → 4 → 5 → 6 → 1

Note: The official test data has been uploaded to Mooshak online judge, but it is hidden from your view. The time limit setting in Mooshak online judge for Subtask A, B, C, D, and E (R-option) are all 1 second (yes, very strict). You are encouraged to generate and post additional test data in Facebook group. Please use `LaborVerifier.java` to verify whether your custom-made test data (for Subtask A+B+C+D) conform with the required specifications.

[6] The different paths may be not unique, but the maximum number of different paths is unique.

Recreational Subtask. Steven is paranoid. After reading about so many recent accidents along Singapore junctions, he now somehow does not like the taxi to cross **more than k junctions** (vertices in Singapore map) along it's shortest path from vertex 0 to vertex 1. That is, the method `Query` has one additional parameter k, i.e. `Query(k)` and $0 \leq k \leq min(20, V)$.

The shortest path for Figure 1: `0` → `3` → `5` → `6` → `4` → `8` → `9` → `10` → `1` with weight 15 minutes uses 9 junctions and is considered invalid according to Steven's paranoid requirement in this subtask.

The valid shortest path (that uses no more than $k = 7$ junctions) is path: `0` → `3` → `4` → `8` → `9` → `10` → `1` with weight 3+6+3+1+3+2 = 18 minutes as shown in Figure 4. This path is 3 minutes longer than the true shortest path without Steven's paranoid restriction but it is now the best answer for this subtask.

Note that the path: `0` → `2` → `7` → `1` is also 'valid' (it uses no more than $k = 7$ junctions) but the total weight 8+8+5 = 21 minutes is longer than the 18 minutes path above.

However, there may be a case that there is no way for Steven and Grace to go from vertex 0 to vertex 1 using no more than k junctions/vertices. For such cases, output -1 instead.

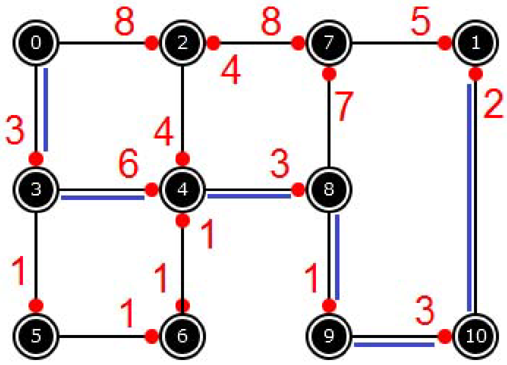

Figure 4: The new valid shortest path on Singapore map as in Figure 1

This Recreational Subtask–which has frustrated your seniors–is not set up in Mooshak this semester and also not graded because of the presence of the new/more challenging? PS5R Subtask E. However, you can freely discuss the idea to solve this variant.